

Erfolgsfaktoren für konsortiale drittmittelfinanzierte Open Source Projekte – am Beispiel von Kitodo 3.0

Matthias Finck

Viele öffentlich geförderten konsortialen Softwareprojekte bringen innovative Ergebnisse hervor, kommen aber zum Projektende nicht über den Status eines Prototyps hinaus und drohen nach Projektende nicht in einen nachhaltigen Nutzungspfad überzugehen. Eine stärkere Zentralisierung des Projektmanagements, eine frühzeitige Überführung der Ergebnisse in nachhaltige Strukturen und vor allem eine hohe Verbindlichkeit in Bezug auf die Auslieferung des Produkts werden in diesem Beitrag als Faktoren zur Erfolgssteigerung am Beispiel der Entwicklung von Kitodo 3.0 vorgestellt.

Many publicly funded consortial software projects produce innovative results, but do not go beyond the status of a prototype after the project ends. Focusing on centralization of project management, early transfer of results into sustainable structures, and above all a high commitment to the delivery of a completely finished product are presented as factors for increasing success in this article.

Einleitung

» Schon seit fast 15 Jahren wird für Bibliotheken bei der Entwicklung von IT-Systemen die „(Wieder-)Verwendung von Komponenten, die von verschiedenen Entwicklern erstellt worden sind und beliebig miteinander kombiniert werden können“¹ gefordert. Open Source Software ist dabei ein wichtiger Baustein, der hilft, dass Softwareentwicklung als gemeinsamer Entwicklungsprozess verstanden und gelebt werden kann². So haben sich mittlerweile mit Systemen wie Kitodo, OPUS, FOLIO u.v.a.m. bibliotheksspezifische Open Source Systeme am Markt etabliert, die vielfach genutzt und von zahlreichen Partnern weiterentwickelt werden.

Nicht nur Bibliotheken sehen die Notwendigkeit gemeinsamer Entwicklungsprozesse. Der Ausschuss für Wissenschaftliche Bibliotheken und Informationssysteme der Deutschen Forschungsgemeinschaft hat 2018 in einem Positionspapier zum Thema „Stärkung des Systems wissenschaftlicher Bibliotheken in Deutschland“ ebenfalls die Bedeutung solch kooperativen Zusammenwirkens bei der Entwicklung von IT-Lösungen betont und explizit bibliothekseigene Entwicklungen in Form von Open Source Systemen neben kommerzielle Angebote gestellt³.

Dabei wird in diesem Papier gefordert, dass „neue Mechanismen, Strukturen oder Organe geschaffen werden, die übergreifende Strategien und Ressourcenplanungen“⁴ ermöglichen – insbesondere in Form von dauerhaften Kooperationen. Auch die Deutsche Initiative für Netzwerkinformationen e.V. (DINI)⁵ beschäftigt sich aktuell mit der Frage der nachhaltigen Finanzierung von IT-Systemen – am Beispiel der Forschungsinfrastrukturen und FDM-Services⁶.

Dabei gibt es inzwischen zahlreiche Beispiele für mehr oder weniger formelle und dauerhafte Entwicklungskooperationen, die die kontinuierliche Entwicklung von gemeinschaftlich getragener Open Source Software sicherstellen⁷. Die Bandbreite reicht von losen Kooperationen mit gemeinsamen Entwicklungsfonds wie das BibApp-Projekt⁸ über konkret getroffene Kooperationsvereinbarungen mit festen Entwicklungsbudgets wie das Projekt beluga core⁹ bis hin zu eigenen Vereinsstrukturen zur Organisation

1 Danowski, P.; Heller, L. (2007): Bibliothek 2.0–Wird alles anders?. In: Bibliothek Forschung und Praxis 31.2, 130-136; S. 131.

2 vgl. Maaß, P. (2016): Free/Libre/Open-Source Software in wissenschaftlichen Bibliotheken in Deutschland. Eine explorative Studie in Form einer Triangulation qualitativer und quantitativer Methoden. Masterthesis, Technische Hochschule Köln, <http://hdl.handle.net/10760/29324> (zuletzt besucht am 01.04.2019).

3 Deutsche Forschungsgemeinschaft (2018): Stärkung des Systems wissenschaftlicher Bibliotheken. In: Information für die Wissenschaft, Nr. 24, https://www.dfg.de/download/pdf/foerderung/programme/lis/180522_awbi_impulspapier.pdf (zuletzt besucht am 01.04.2019)

4 ebda, S. 20.

5 <https://dini.de/>

6 https://www.forschungsdaten.org/index.php/Wer_soll_das_bezahlen%3F_Kosten-_und_Betriebsmodelle_f%C3%BCr_nachhaltige_Forschungsinfrastrukturen_und_FDM-Services

7 vgl. Finck, M. (2016): Klein aber oho! Einfache kooperative Softwareentwicklung in Bibliotheken – am Beispiel von beluga core. In: b.i.t.online, 4/19, Verlag Dinges & Frick, S. 315-320.

8 <https://verbundwiki.gbv.de/display/VZG/BibApp>

9 <https://www.beluga-core.de/>

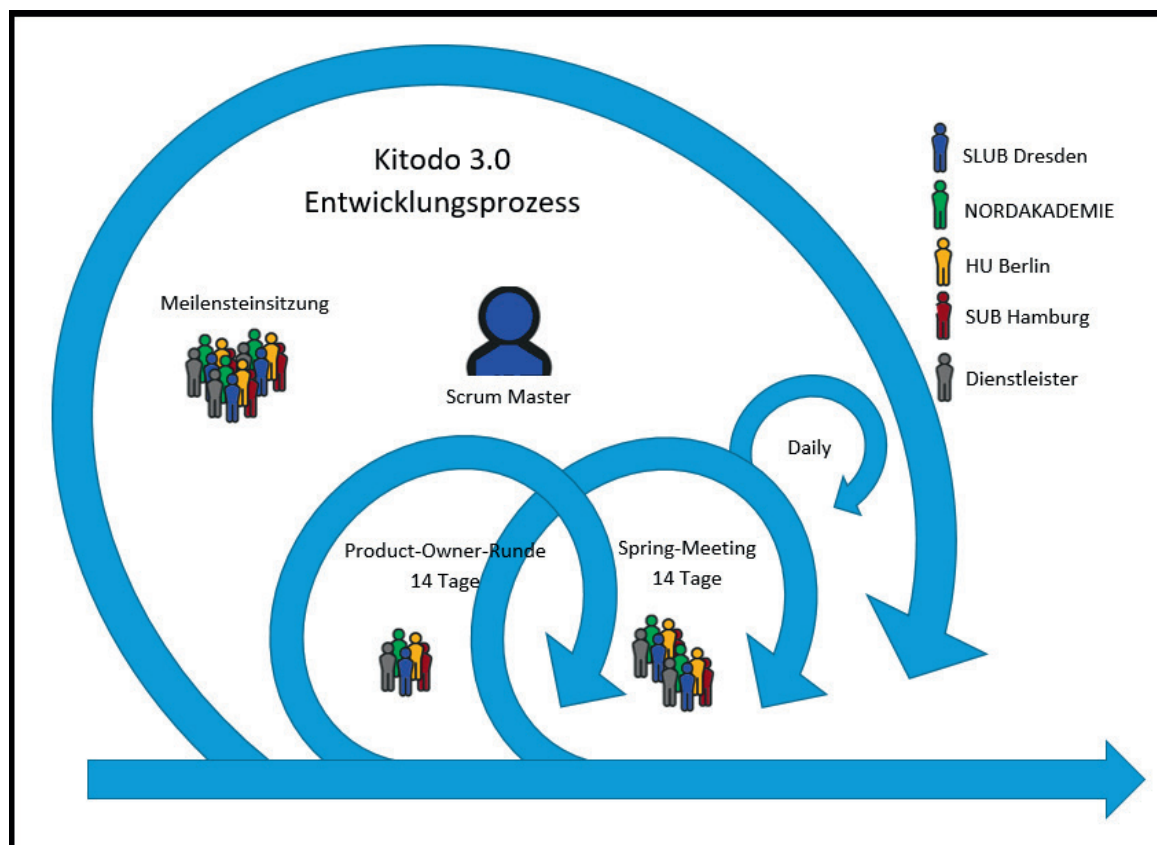


Abb. 1:
Entwicklungs-
prozess Kitodo 3.0

des gemeinsamen Entwicklungsprozesses wie beim Projekt Kitodo¹⁰.

All diesen Kooperationsformen ist eines gemeinsam: Die Budgets für die Entwicklung dienen der kontinuierlichen Weiterentwicklung und Wartung des Systems. Große Innovationssprünge oder technologische Erneuerungsprozesse sind mit diesen eher kleinen Budgets selten leistbar. Diese großen Entwicklungsschritte erfolgen entweder durch das finanzielle bzw. personelle Engagement einzelner Bibliotheken bzw. Dienstleister oder sind Teil eines gemeinsamen Entwicklungsvorhabens – häufig gefördert durch Drittmittelgeber wie die DFG.

Leider ist bei diesen großen konsortialen, drittmittelfinanzierten Vorhaben allzu oft zu beobachten, dass die Ergebnisse nicht über den Status eines Prototyps hinausgehen und dementsprechend oftmals nicht in langfristigen Entwicklungsprozessen münden. In diesem Beitrag sollen mögliche Kriterien für nachhaltige erfolgreiche, konsortiale, drittmittelfinanzierte Open Source Projekte am Beispiel der DFG-geförderten Entwicklung von Kitodo 3.0 vorgestellt werden.

Der Entwicklungsprozess von Kitodo 3.0

Kitodo ist eine Open Source Software zur Unterstützung der Digitalisierung von Kulturgut in Bibliothe-

ken, Archiven, Museen und Dokumentationszentren. Das Workflowmanagement-Modul Kitodo.Production wird derzeit noch bis Ende Juni 2019 mit einer Förderung der Deutschen Forschungsgemeinschaft (DFG) weiterentwickelt. Dabei handelte es sich um ein konsortiales Projekt mit einer Laufzeit von knapp drei Jahren, an dem die Sächsische Landesbibliothek – Staats- und Universitätsbibliothek Dresden (SLUB), die Staats- und Universitätsbibliothek Hamburg (SUB), die Humboldt-Universität zu Berlin (HU) und die NORDAKADEMIE, Hochschule der Wirtschaft Elmshorn (NAK), direkt als Projektpartner beteiligt waren und einen Großteil des Entwicklungsteams gestellt haben. Ergänzt wurde dieses Team durch Mitarbeiter der beiden Kitodo-Dienstleister Zeutschel und effective WEBWORK. Insgesamt bestand das Team somit aus ca. 15 Personen, von denen gut die Hälfte auch unmittelbar an der Implementierung beteiligt gewesen sind.

Das Projekt verfolgte drei große Entwicklungsziele: Das zentrale Entwicklungsziel war eine konsequente Modularisierung der Software, wodurch die Voraussetzungen für zukünftige Erweiterungen und Anpassungen geschaffen werden sollen. Hierzu wurden große Bestandteile des technischen Kerns der Software ausgetauscht. Es handelte sich also um einen

¹⁰ <http://www.kitodo.org/>

grundsätzlichen technischen Relaunch der Software. Daneben wurde als zweites Entwicklungsziel die Benutzeroberfläche gemäß den heutigen Anforderungen an Usability und unter Nutzung aktueller technischer Möglichkeiten grundlegend überarbeitet. Die grafische Oberfläche hat ein komplettes Re-Design mit dem Ziel einer insgesamt verbesserten und intuitiveren Benutzerführung erfahren. Als drittes Ziel sollte die Software für weitere Medienarten und -formate geöffnet und damit stärker flexibilisiert werden.

Der Entwicklungsprozess wurde dabei als agiler Prozess in Anlehnung an die Entwicklungsmethodik SCRUM gestaltet. Die Projektleiter der vier Standorte bildeten die sogenannte Product-Owner-Runde, die sich im zweiwöchigen Rhythmus traf, um die klassischen Aufgaben der Rolle eines Product Owners auszufüllen: Die Runde entwickelte die Produktvision und vertrat diese nach außen. Außerdem wurden in der Runde – ganz der Rolle entsprechend – die Ideen entwickelt, wie das fertige Produkt aussehen könnte, wer es wann mit welchem Ziel nutzen wird, wie es mit bestehenden Prozessen interagiert, und vor allem wie daraus ein Mehrwert für die Community entsteht¹¹.

Neben der Product-Owner-Gruppe fanden ebenfalls im zweiwöchigen Rhythmus sogenannte Sprintsitzungen des Entwicklungsteams statt, in denen der SCRUM-Master als Vertreter des Konsortialführers aus Dresden mit dem Entwicklungsteam die Aufgaben des nächsten Entwicklungssprints abgestimmt hat. Diese beiden Abstimmungs- und Planungsunden fanden verzahnt im wöchentlichen Wechsel statt. Ergänzt wurden diese beiden Runden durch tägliche Entwickler-MeetUps (Daily Scrum Meetings) und ca. quartalsweise stattfindenden Meilensteinsitzungen des Gesamtprojektteams.

Mit Ausnahme der Meilensteinsitzungen fanden alle anderen Runden aufgrund der hohen räumlichen Verteiltheit des Teams virtuell in Form von Videokonferenzen statt.

Grundsätzliche Projektrisiken konsortialer, drittmittelfinanzierter Entwicklungen

Aufgrund des großen Projektkonsortiums und der hohen Verteiltheit – nicht nur räumlichen Verteiltheit des Entwicklungsteams, sondern vor allem auch der verteilten Verantwortlichkeiten und Ziele der Projektpartner – bot die Projektkonstellation eine Reihe typischer Risiken, die oftmals zum Scheitern solcher konsortialen, drittmittelgeförderten Entwicklungsprojekte führen.

Ein zentrales Risiko stellen die verbindlichen Individualziele eines jeden Konsortialpartners gegenüber dem Auftraggeber dar. So gab es auch in diesem Projekt klare inhaltliche Abgrenzungen, wonach z.B. die SLUB Dresden sich verpflichtet hatte, sich hauptsächlich um die Neugestaltung der technischen Architektur zu kümmern (Projektziel 1), die Standorte in Hamburg und Elmshorn vorwiegend dem Ziel der Neugestaltung der Benutzungsschnittstelle verpflichtet waren (Projektziel 2) und die HU Berlin sich u.a. mit der Flexibilisierung der Formate zu beschäftigen hatte (Projektziel 3). Diese verteilte Verantwortung für bestimmte Aspekte des Gesamtprojekts birgt systemisch das Risiko, dass jeder Partner konsequent seine Projektziele verfolgt und – im Idealfall – auch zu guten Ergebnissen kommt, dass diese Teilergebnisse aber nicht genügend aufeinander abgestimmt sind, als dass sie ein in sich stimmiges Ganzes ergeben. Und das ist bei dem Projektziel einer gemeinsamen Softwaregestaltung natürlich besonders kritisch.

Ein weiteres grundsätzliches Risiko, das bei dieser Konstellation eines Softwareprojekts verstärkt auftritt, stellt das Pareto-Prinzip dar, das auch in der Softwareentwicklung gilt und wonach die letzten 20% zur Fertigstellung einer Software 80% des Entwicklungsaufwandes bedeuten¹². Die Zielformulierung in erfolgreichen eingeworbenen Drittmitteln basiert in der Regel auf Innovationen und Neuerungen. Und um den Zielen gerecht zu werden, wird ein Großteil der Entwicklungsarbeit in diese Innovationen investiert. Das birgt das Risiko, dass die konzeptionellen Ergebnisse zwar hervorragend sein können, die Implementierung aber eben nicht über einen prototypischen Status hinauskommt, weil am Ende die Ressourcen fehlen, die letzten 20% zur Produktreife mit dem dafür notwendigen hohen Ressourcenbedarf auch umzusetzen. Das Problem verstärkt sich bei öffentlichen, drittmittelfinanzierten Projekten noch, weil der Erfolg der handelnden Personen nicht in dem kommerziell erfolgreichen Einsatz der Software gemessen wird, sondern an dem im Projektantrag formulierten Erkenntnisgewinn. Normalerweise sind keine unmittelbaren persönlichen und wirtschaftlichen Konsequenzen zu befürchten, wenn sich bei dieser Art von drittmittelfinanzierten Projekten kein nachhaltiger Nutzungserfolg einstellt. Der wirtschaftliche Erfolgsdruck ist deutlich geringer.

Ein drittes Risiko ist der Übergang von der Entwicklung in die langfristige Pflege und Wartung der entwickelten Software. Selbstverständlich weisen die

¹¹ vgl. Pichler, R. (2014): Agile Produktmanagement mit Scrum: Erfolgreich als Product Owner arbeiten. Dpunkt. Verlag.

¹² Koch, R. (2006). Das 80/20-Prinzip. Mehr Erfolg mit weniger Aufwand. In: Das Summa Summarum des Erfolgs (S. 249-263). Gabler.

allermeisten drittmittelfinanzierten Projekte einen Plan aus, wie die Software nach Projektende nachhaltig gepflegt werden soll. Und zumeist gibt es auch funktionierende Finanzierungsmodelle, die die Pflege- und Wartungskosten berücksichtigen – siehe die einleitenden Beispiele von dauerhaften Entwicklungskooperationen. Problematisch wird es aber, wenn die Software zum geplanten Projektende eben nicht den vollen Produktstatus erreicht hat – und das ist bei Softwareentwicklungsprojekten unabhängig von der Art der Finanzierung oftmals der Fall. Größere Studien können zeigen, dass bis heute „nur rund ein gutes Viertel aller Projekte erwartungs- und plangemäß vollendet wird.“¹³. Anders als in industrieller Softwareentwicklung, wo dann der Hersteller im Zweifel nachträgliche Mittel zur Fertigstellung zur Verfügung stellen muss, ist diese nachträgliche Finanzierung bei drittmittelfinanzierten Projekten fast ausgeschlossen. Das erhöht das Risiko, dass es zu einer Finanzierungslücke nach Projektende kommt, die es erschwert, die notwendigen Abschlussarbeiten vornehmen zu können.

Die drei zentralen Erfolgsfaktoren im Kitodo 3.0 Projekt

Im Rahmen des Bibliothekskongresses 2019 in Leipzig wurde eine Beta-Version von Kitodo 3.0 vorgestellt, die schon eine gewisse Produktreife aufweist und das Team sehr optimistisch im Hinblick auf die finale Version 3.0 zum Projektende blicken lässt.

Bei der Frage, warum nun ausgerechnet in diesem Projekt zu klappen scheint, was vielfach misslingt, lassen sich drei Faktoren ausmachen, die die Erfolgswahrscheinlichkeit haben deutlich steigen lassen:

1. Starke Zentralisierung des Projektmanagements im Entwicklungsteam: Um das Risiko der Verzettlung in Individualzielen zu reduzieren, wurde das Gremium der Product-Owner-Runde ins Leben gerufen. Hier haben alle Teilprojektleiter sich in engen zweiwöchigen Rhythmen auf eine gemeinsame Entwicklungslinie und Priorisierung verständigt. Außerdem wurde der Rolle des Scrummasters eine hohe Macht zugesprochen. Trotz der Verteiltheit des Projekts durfte der Scrummaster über die personellen Ressourcen in den Sprintsitzungen standortübergreifend verfügen. Das hat einen einheitlichen Entwicklungsplan ermöglicht. Die Konsortialpartner sind bei dieser Konstellation bewusst das Risiko eingegangen, dass ihre Individualziele wie z.B. die Flexibilisierung der Formate nicht so umfassend umgesetzt werden wie bei einer Konzentration der eigenen Ressourcen

auf diese Fragestellung. Dafür stieg die Wahrscheinlichkeit am Ende eine vollständige Softwarelösung zur Verfügung stellen zu können.

2. Hohe Verbindlichkeit in der Lieferung: In der Gesamtprojektkonstellation ergab sich relativ früh die Chance, die Software bei einem konkreten Kunden in die Anwendung zu bringen. Das Bundesarchiv der Schweiz plante die Version Kitodo 3.0 ab April 2019 zum Einsatz zu bringen. Und Teile des Entwicklungsteams aus dem DFG-Projekt waren ebenfalls für die termingerechte Auslieferung beim Bundesarchiv verantwortlich. Diese Zusage von Teilen der Entwicklung für ein frühzeitiges und unverschiebbares Einsatzszenario sorgte beim Entwicklungsteam für eine hohe Awareness bezüglich des Pareto-Prinzips. Die Aufgaben zur abschließenden Produktentwicklung wurden rechtzeitig in dem vorhandenen Projektbudget eingeplant.

3. Überführung in etablierte Strukturen: Hinter der Software von Kitodo steht eine Community, die sich durch einen Verein organisiert. Dieser Verein hat ein halbes Jahr vor Projektende konkrete Pläne

Anzeige



Lösungen für Bibliotheken & Archive

aDIS/BMS – das integrierte Bibliothekssystem

- | für alle Bibliothekstypen: öffentliche, wissenschaftliche und Archive
- | auch als cloudfähiges Mandanten-, Verbundsystem oder als Lokalsystem in Verbänden

aDIS/OPAC



- | barrierefrei & responsive Design
- | Autocomplete & Facettierung
- | als Discovery-System, auch mit externen Indices
- | e-Payment, Merklisten u.v.m.
- | Single Sign-on

aDIS/Wissen – das integrierte CMS, z.B. für die OPAC-Gestaltung oder Neuerwerbungslisten

aStec-Datenservice – der Fremddaten-Pool für Kunden, z.B. mit angereicherten DNB-, EKZ-Daten

systematisch anders

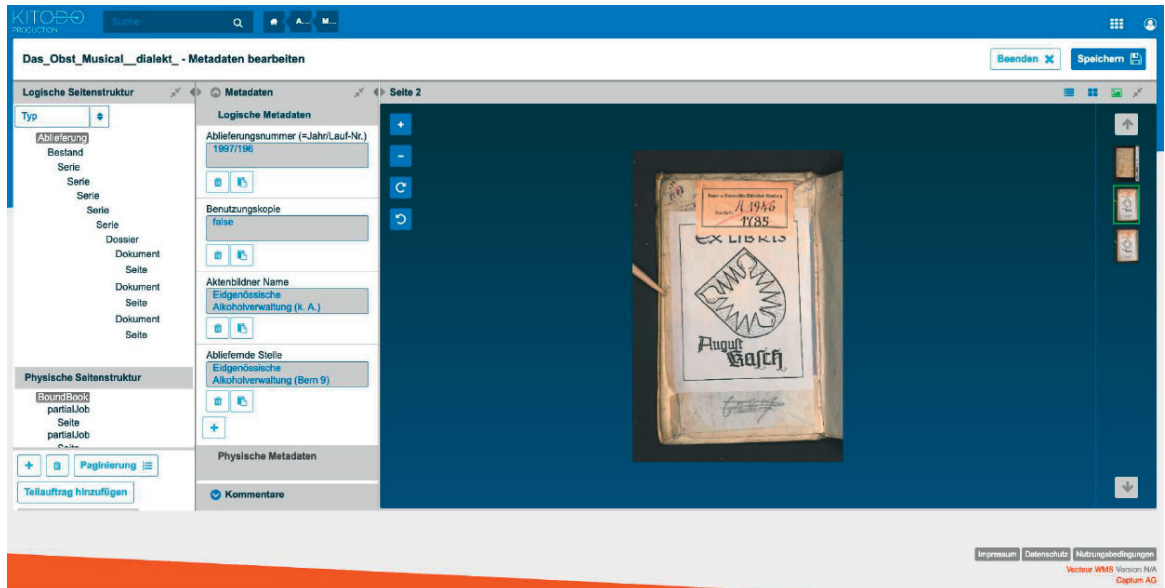
systematisch gut

aStec eG
Paul-Lincke-Ufer 7c
10999 Berlin
Tel. 030/617 939-0
info@astec.de

www.astec.de

13 Brödner, P. (2007). Der (faule) Zauber der IT. Arbeit, Innovation und Nachhaltigkeit Band 7, 27; S. 30.

Abb. 2:
Screenshot von
Kitodo 3.0



entwickelt, wie die Software nach Projektende wieder durch die Vereinsstrukturen weitergeführt wird. Außerdem war im Verein das Bewusstsein vorhanden, dass in der ersten Phase der Bereitstellung einer so grundlegend überarbeiteten Software erhebliche zusätzliche Mittel notwendig sind zur Migrationsunterstützung der Vereinsmitglieder auf die neue Software sowie abschließenden Entwicklungsbedarf. Dadurch stehen nicht nur die üblichen Budgets für eine kontinuierliche Pflege und Wartung der Software zur Verfügung, sondern zusätzliche Mittel für den Entwicklungsprozess, sodass dieser langsam zurückgefahren werden kann.

Der Verein der Kitodo-Community ist zweifelsohne ein wichtiges Bindeglied zwischen Projektentwicklung und kontinuierlicher Weiterentwicklung und damit ein Erfolgsfaktor für die nachhaltige Weiterführung der Projektergebnisse. Trotzdem wäre auch der Verein überfordert, würde die Software nicht am Ende der Projektphase schon annähernd produktreif erlangt haben. Insofern scheinen die beiden anderen Faktoren – Zentralisierung des Teams und hohe Verbindlichkeit – noch wichtiger für den Projekterfolg. Während der Entwicklung des letzten halben Jahres war offensichtlich, dass die festen Zusagen dem Bun-

desarchiv gegenüber das Entwicklungsteam, aber auch die Product-Owner in ihren Wünschen und Forderungen an zusätzliche Funktionalität stark diszipliniert haben. Es gelang durch äußere Faktoren, einen Erfolgsdruck aufzubauen, der an der Einsatzfähigkeit der Software gemessen wurde und dem sich alle im Konsortium untergeordnet haben.

Außerdem gab es mehrmals im Projekt die Gefahr, dass die Entwicklungsstränge der einzelnen Projektpartner auseinanderzulaufen drohten. Hier war die große Entscheidungsbefugnis, die dem Scrummaster eingeräumt wurde, absolut notwendig, um diesen Tendenzen rechtzeitig entgegenzuwirken.

Selbstverständlich sind dieses drei Faktoren keine generelle Erfolgsgarantie für diese Form von Projekten. Aber die Berücksichtigung dieser Faktoren scheint zumindest hilfreich zu sein und ist so vielleicht eine nutzbringende Erkenntnis für zukünftige Projekte. Deshalb sollten sich die Beteiligten solcher Entwicklungsprojekte genau überlegen, wie ein Projektumfeld geschaffen werden kann, in dem diese Faktoren möglichst stark zur Geltung kommen. Das ist im Einzelfall sicherlich nicht einfach, aber das Ergebnis lohnt sich. ■



Prof. Dr. Matthias Finck

Der promovierte Informatiker hat einen Lehrstuhl für Usability Engineering an der Nordakademie und ist Inhaber der effective WEBWORK GmbH.
matthias.finck@nordakademie.de
finck@effective-webwork.de